

# Optimal control for a robotic exploration, pick-up and delivery problem

Vladislav Nenchev, Christos G. Cassandras, and Jörg Raisch

**Abstract**—This paper addresses an optimal control problem for a robot that has to find and collect a finite number of objects and move them to a depot in minimum time. The robot has fourth-order dynamics that change instantaneously at any pick-up or drop-off of an object. The objects are modeled by point masses with a-priori unknown locations in a bounded two-dimensional space that may contain unknown obstacles. For this hybrid system, an Optimal Control Problem (OCP) is approximately solved by a receding horizon scheme, where the derived lower bound for the cost-to-go is evaluated for the worst and for a probabilistic case, assuming a uniform distribution of the objects. First, a time-driven approximate solution based on time and position space discretization and mixed integer programming is presented. Due to the high computational cost of this solution, an alternative event-driven approximate approach based on a suitable motion parameterization and gradient-based optimization is proposed. The solutions are compared in a numerical example, suggesting that the latter approach offers a significant computational advantage while yielding similar qualitative results compared to the former. The methods are particularly relevant for various robotic applications like automated cleaning, search and rescue, harvesting or manufacturing.

**Index Terms**—Optimal control, hybrid systems, motion control.

## I. INTRODUCTION

One of the major challenges in autonomous robotic navigation is coping with uncertainties arising from limited a-priori knowledge of the environment. Acquiring necessary information and achieving the overall goal are complementary subtasks that require adapting the motion of a robot during mission execution, typically accompanied by minimizing a performance criterion. In this work we address an Optimal Control Problem (OCP) for a robot with fourth-order dynamics that has to find, collect and move a finite number of objects to a designated spot in minimum time. The objects with a-priori known masses are located in a bounded two-dimensional space, where the robot is capable of localizing itself using a state-of-the-art simultaneous localization and mapping (SLAM) system [1]. The challenging aspects of the problem at hand are (at least) threefold. One of them arises due to the discontinuity of the value function denoting

the overall completion time, which makes it hard to obtain an explicit controller even for deterministic linear systems [2], [3]. Fortunately, a wide range of approximate solutions has been proposed, including approaches based on numerical continuation [4], value set approximation [5], multi-parametric programming [6] etc. Another challenge follows from the requirement to collect a finite number of objects and drop them at a particular spot, both leading to autonomous switchings of the robot's continuous dynamics. While deterministic versions of this problem can be handled efficiently, e.g., by two-stage optimization [7], [8] or relaxation [9], the complexity of most approaches for stochastic setups scales poorly with the problem size [10]. Since the robot has to reach the corresponding locations of the objects or the depot with minimal overall cost, the overall problem also contains an instance of the well-known NP-hard Traveling Salesperson Problem (TSP) [11]. Further, optimal exploration of a limited space is an inherently difficult problem by itself. Minimizing the expected time for detecting a target located on a real line with a known probability distribution by a searcher that can change its motion direction instantaneously, has a bounded maximal velocity and starts at the origin, was originally addressed in [12]. Different versions of this problem have received considerable attention from several research communities, e.g., as a “pursuit-evasion game” in game theory [13], [14], as a “cow-path problem” in computer science [15] or as a “coverage problem” in control [16], [17], but its solution for a general probability distribution or a general geometry of the region is, to a large extent, still an open question. Effective approaches for the related persistent monitoring problem based on estimation [18], linear programming [19] or parametric optimization [20] have been also been proposed. OCPs with uncertainties have also been addressed by certainty equivalent event-triggered [21], minimax [22] and sampling-based [23] optimization schemes. While methods for Partially Observable Markov Decision Processes (POMDP's) can also be applied, e.g., [24], [25], they typically become computationally infeasible for larger problem instances. Due to the aforementioned aspects, the problem at hand has exponential complexity in the number of objects and for any chosen time and space discretization. In this context, employing a discrete abstraction of the underlying continuous dynamics is often only possible by introducing a hierarchical decomposition [26], or additional assumptions that simplify the implementation of automatically synthesized hybrid controllers [27]. Alternatively, one may resort to receding horizon approaches that have been shown to outperform other optimization methods under the presence of uncertainty, e.g., for the elevator dispatching problem [28],

V.Nenchev is with the Control Systems Group, Technische Universität Berlin, Germany. Corresponding email: nenchev@control.tu-berlin.de

C.G.Cassandras is with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, MA, USA. cgc@bu.edu

J.Raisch is with the Control Systems Group, Technische Universität Berlin and the Systems and Control Theory Group, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany. raisch@control.tu-berlin.de

C.G.Cassandras was supported in part by NSF under grants CNS-1239021, ECCS-1509084, and IIP-1430145, by AFOSR under grant FA9550-15-1-0471, and by ONR under grant N00014-09-1-1051.

multi-agent reward collection problems [29] or planning with temporal logic constraints [30].

For a scenario where the number of objects is finite but unknown, a combined optimal exploration and control scheme for a robot that has to find, collect and move objects in a two-dimensional position space was proposed in [31]. The approach was based on a policy enforcing a pick-up upon an object's detection, followed by a certainty equivalent discrete optimization on a finite abstraction of the robot's motion in the environment. This heuristic restriction was omitted in [32], where optimal exploration and control solutions for the worst and a probabilistic case assuming a uniform distribution of the objects on a line interval were derived. Since a direct generalization of this result for higher dimensions was not possible, this paper proposes and compares two approximate receding horizon approaches. The first is based on discretizing time and space and solving a non-convex OCP over a finite horizon by a Mixed Integer Programming (MIP) implementation. In the second approach, the motion of the robot is parameterized by a finite number of parameters. This enables the use of Infinitesimal Perturbation Analysis (IPA) [33] to solve the worst and probabilistic case OCPs by a bi-level iterative optimization scheme, solved only whenever new information becomes available. Preliminary versions of these approaches along a fixed exploratory trajectory have been presented in [34]. Here we extend the methods such that the shape of the exploratory trajectory can be adjusted online, which is particularly useful under the presence of a-priori unknown obstacles.

The remainder of the paper is organized as follows: in Sec. II, we present the problem formulation. Sec. III starts with a brief discussion on the performance index and introduces a lower bound for the cost-to-go, followed by the proposed time-driven (Sec. IV) and event-driven approaches (Sec. V). The four methods are then compared in a numerical example (Sec. VI), followed by the conclusions in Sec. VII.

**Notation.** For a set  $S$ ,  $|S|$  and  $2^S$  denote its cardinality and the set of all of its subsets (power set), respectively. For  $r \in \mathbb{R}$ , respectively,  $r \in \mathbb{R}^n$ ,  $|r|$  and  $\|r\|$  denote the absolute value and the Euclidean norm.  $\mathbf{I}_n$  is an identity matrix with dimension  $n$ .  $\mathbf{0}_{m,n}$  represents an  $m \times n$  matrix with zero entries. For a vector of zeros or ones with length  $m$ , we write  $\mathbf{0}_m$  or  $\mathbf{1}_m$ , respectively.  $\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{> 0}$  denote the sets of reals, non-negative reals and positive reals, respectively. We use the derivatives  $\dot{x}(t) = \frac{dx(t)}{dt}$ ,  $c'(s, \theta) = \frac{\partial c(s, \theta)}{\partial s}$  and the gradient  $\nabla_{\theta} c(s, \theta) = \left[ \frac{\partial c(s, \theta)}{\partial \theta_1}, \dots, \frac{\partial c(s, \theta)}{\partial \theta_n} \right]^T$ .

## II. PROBLEM FORMULATION

Consider a finite set of objects  $O = \{o_1, \dots, o_L\}$ , where every  $o_l, l \in \{1, \dots, L\}$ , is uniquely characterized by its position  $p^{(l)} \in \mathcal{Y}_g$ ,  $\mathcal{Y}_g = [-y_{\max}, y_{\max}] \times [-y_{\max}, y_{\max}] \subset \mathbb{R}^2$ , and mass  $m^{(l)} \in \mathbb{R}_{\geq 0}$ . A robot has to find, collect and move all objects back to a designated spot (depot), located at  $y_d = \mathbf{0}_2$ , in minimum time. The robot is equipped with an

omni-directional sensor footprint of size  $r \ll y_{\max}$  around its current position  $y(t) \in \mathbb{R}^2$ , hence covering the area

$$\mathcal{O}(y(t)) = \{y_p \in \mathbb{R}^2 : \|y(t) - y_p\| \leq r\}. \quad (1)$$

The overall system is modeled by a hybrid automaton [35], i.e., a 9-tuple  $\mathcal{H} = \{Q, \mathcal{X}, F, U, E, \text{Inv}, G, R, \text{Init}\}$ . The discrete state at time  $t$  is  $q(t) = (q_1(t), q_2(t), q_3(t))$ , where  $q_1(t) \subseteq O$  is the set of objects being carried by the robot,  $q_2(t) \subseteq O$  the set of objects that has been dropped at the depot prior to or at time  $t$ , and  $q_3(t) \subseteq O$  is the set of objects that have been detected so far. Clearly,  $q(t) \in Q$  with  $Q \subseteq 2^O \times 2^O \times 2^O$ . The current mass of the robot is  $m_q(t) = m_0 + \sum_{l, o_l \in q_1} m^{(l)}$ , where  $m_0$  is the nominal mass of the robot. The overall continuous state  $(x(t), \mathcal{Y}(t)) \in \mathcal{X}$  consists of the robot state  $x(t) = [y^T(t) \ v^T(t)]^T \in X$ , where  $v(t) \in \mathbb{R}^2$  is the current velocity of the robot, and the region  $\mathcal{Y}(t) \subseteq \mathcal{Y}_g$  that has not been explored at time  $t$ . The robot state  $x(t)$  evolves according to a finite collection of vector fields  $F = \{f_q\}_{q \in Q}$ , i.e.

$$\dot{x}(t) = f_q(x, u) = \begin{bmatrix} \mathbf{0}_{2,2} & \mathbf{I}_2 \\ \mathbf{0}_{2,2} & \mathbf{0}_{2,2} \end{bmatrix} x(t) + \frac{1}{m_q(t)} \begin{bmatrix} \mathbf{0}_{2,2} \\ \mathbf{I}_2 \end{bmatrix} u(t), \quad (2)$$

driven by the piecewise continuous control signal  $u : [0, t_f] \rightarrow U := \{\phi \in \mathbb{R}^2 : \|\phi\| \leq 1\}$ , where  $t_f$  is the free final time for the overall assignment. As  $Q$  is finite, the set of discrete state transitions (or events)  $E \subseteq Q \times Q$  is also finite. Let  $E$  be partitioned into  $\Delta \cup \Pi \cup \Psi$ , where for  $q = (q_1, q_2, q_3)$ ,  $q' = (q'_1, q'_2, q'_3) \in Q$ ,

$$\Delta = \{(q, q') : q'_1 = q_1, q'_2 = q_2, q'_3 = q_3 \cup \{o_l\}, o_l \notin q_3\}$$

is the set of detection events,

$$\Pi = \{(q, q') : q'_1 \setminus q_1 = \{o_l\}, o_l \in q_3, q'_2 = q_2, q'_3 = q_3\},$$

is the set of pick-up events, and

$$\Psi = \{(q, q') : q_1 \neq \emptyset, q'_1 = \emptyset, q'_2 = q_2 \cup q_1, q'_3 = q_3\}$$

corresponds to the set of drop-off events. With the introduced sensor paradigm (1), detection events occur when the distance between the current robot position and the position of an object that has not been detected so far becomes  $r$ . Pick-up events occur when the robot reaches the position of an object that has not been collected so far. Drop-off events occur when the robot reaches the depot and carries objects. In addition, for both pick-up and drop-off events, zero velocity is required. The corresponding conditions on  $q$  and  $x$  for the occurrence of detection, pick-up and drop-off events are captured by the invariant  $\text{Inv} : Q \rightarrow 2^X$ , i.e.,

$$\text{Inv}(q) = \begin{cases} X \setminus \{[y^T \ v^T]^T : \|y - p^{(l)}\| \neq r\}, & \text{if } o_l \notin q_3, \\ X \setminus \{[p^{(l)T} \ \mathbf{0}_2^T]^T\}, & \text{if } o_l \notin q_1 \cup q_2, \\ X \setminus \{\mathbf{0}_4\}, & \text{if } q_1 \neq \emptyset, \end{cases}$$

and the guard map  $G : E \rightarrow 2^X$ , i.e. with  $e = (q, q')$ ,

$$G(e) = \begin{cases} \{[y^T \ v^T]^T : \|y - p^{(l)}\| = r\}, & \text{if } e \in \Delta, q'_3 \setminus q_3 = \{o_l\}, \\ \{[p^{(l)T} \ \mathbf{0}_2^T]^T\}, & \text{if } e \in \Pi, q'_1 \setminus q_1 = \{o_l\}, \\ \{\mathbf{0}_4\}, & \text{if } e \in \Psi, q_1 \neq \emptyset. \end{cases}$$

For example, upon a detection of a new object as per (1), when the robot is in the discrete state  $q$ , the first case of the Inv requires that a transition must occur, and the first case of  $G$  allows a transition to a discrete state  $q'$ , where the discovered object is included in the detected objects set  $q'_3$ . The reset map  $R : E \times X \rightarrow 2^X$  is trivial since no jumps of the continuous variables occur upon a discrete state switching. Note that the above conditions do not depend on  $\mathcal{Y}(t)$ , and hence, Inv,  $G$  and  $R$  map into  $2^X$  instead of  $2^{\mathcal{X}}$ . As both the robot and the objects are represented by points in  $\mathcal{Y}_g$ , we assume that no collisions can occur. A practical setup that satisfies this assumption is, e.g., a quadrotor that has to explore a two-dimensional space on the ground from above. Finally, as the robot is assumed to start at the depot with zero velocity, and no objects have been detected, picked up or dropped off before that, the initial state set is  $\text{Init} = \{(q(0), (x(0), \mathcal{Y}(0)))\} = \{((\emptyset, \emptyset, \emptyset), (\mathbf{0}_4, \mathcal{Y}_g \setminus \mathcal{O}(\mathbf{0}_2)))\}$ .

**Remark 1.** Obstacles in  $\mathcal{Y}_g$  can be easily included in the proposed approaches. However, to keep notation as simple as possible, we omit their presence in the main analysis and briefly outline the solution that was used to handle the obstacle in the numerical example (Sec. VI) in a follow-up remark.

Solving the addressed problem involves  $L$  detection events,  $L$  pick-up events and up to  $L$  drop-off events, as it can be advantageous to collect several objects on the way and drop them off simultaneously at the depot. Hence, for the total number  $N$  of events,  $2L < N \leq 3L$  holds. The time of the occurrence of event  $n$ ,  $1 \leq n \leq N$  is denoted by  $t_n$ ,  $t_0$  is the initial time,  $t_f = t_N$  the final time, and  $t_0 \leq t_1 \leq \dots \leq t_N$ . The  $N$  time intervals  $\tau_n := [t_{n-1}, t_n]$ ,  $n = 1, \dots, N$  form the time axis from the initial to the final time with  $\tau := (\tau_1, \dots, \tau_N)$ . The input is an ordered set of functions  $u = (u^1, \dots, u^N)$ , where  $u^n : \tau_n \rightarrow U$  are absolutely continuous functions for  $n \in \{1, \dots, N\}$ . Thus, if  $\zeta = (\tau, q, \xi)_u$  is an execution of the hybrid automaton  $\mathcal{H}$  for an input signal  $u$ , i.e.  $(\tau, q, \xi)_u \models \mathcal{H}$ ,  $q = (q^1, \dots, q^N)$  is a discrete state trajectory with  $q^n : \tau_n \rightarrow Q$ ,  $q^n = \text{const}$ .  $\xi = (\xi^1, \dots, \xi^N)$  is the continuous state trajectory with  $\xi^n = (x^n, \mathcal{Y}^n)$ , where  $x^n : \tau_n \rightarrow X$  are absolutely continuous functions, and  $\mathcal{Y}^n : \tau_n \rightarrow 2^{\mathcal{Y}_g}$  non-increasing functions, i.e.,  $\mathcal{Y}^n(t') \subseteq \mathcal{Y}^n(t)$  for  $t \leq t'$ . The cost of an execution is the total task time

$$t_f = \sum_{n=1}^N (t_n - t_{n-1}) = t_N - t_0. \quad (3)$$

Let  $\text{Fin} = \{(q_f = (\emptyset, O, O), (\mathbf{0}_4, \mathcal{Y}_g \setminus \cup_{\tilde{t} \in [0, t_f]} \mathcal{O}(y(\tilde{t}))))\}$  denote the set of states that can be reached upon completing the task. One way to account for the uncertainty in the addressed OCP is to minimize, at time  $t$ , the largest cost that

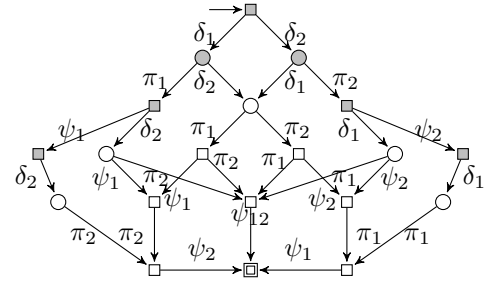


Fig. 1. Discrete dynamics of  $\mathcal{H}$  for  $O=\{o_1, o_2\}$  with  $\delta_l \in \Delta$ ,  $\pi_l \in \Pi$ ,  $\psi_l \in \Psi$ ,  $l \in \{1, 2\}$ ,  $\psi_{12} \in \Psi$ . Exploration takes place at gray states. The robot has zero velocity at states denoted by a square.

may occur for a possible configuration of all objects that have not been discovered so far. Alternatively, the positions of the objects that have not been detected so far can be assumed to be independent identically distributed random variables with probability density functions

$$\mathcal{P}(p^{(l)}) = \begin{cases} \frac{1}{\kappa(t)}, & \text{if } p^{(l)} \in \mathcal{Y}(t), \\ 0, & \text{if } p^{(l)} \in \mathcal{Y}_g \setminus \mathcal{Y}(t), \end{cases} \quad (4)$$

$\forall l$ , where  $\kappa(t)$  measures the size of  $\mathcal{Y}(t)$ . This leads to the following worst-case (A) and probabilistic (B) OCPs.

**Problem 1.** At state  $(q(t), \xi(t))$ , find the input signal  $u|_{[t, t_f]}$  for  $\mathcal{H}$ , such that for  $p = \{p^{(l)} : o_l \notin q_3(t)\}$

- A)  $\min_{u|_{[t, t_f]}} \max_p (t_f - t)$ , s.t.  $(q(t_f), (x(t_f), \mathcal{Y}(t_f))) \in \text{Fin}$ ;
- B)  $\min_{u|_{[t, t_f]}} E\{t_f - t\}$ , s.t.  $(q(t_f), (x(t_f), \mathcal{Y}(t_f))) \in \text{Fin}$ .

Note that Problem A is always deterministic, while Problem B is probabilistic until the last detection of an object.

The outline of the solution reads as follows. First, we provide a discussion on the time-optimal value function and derive a lower bound for the cost-to-go. Then, we propose two approximation-based approaches for Problems A and B – one that requires time discretization and re-computation at every time step, and one based on motion parameterization that allows for an event-driven implementation, i.e., the corresponding OCPs are re-solved only upon the occurrence of a detection event.

### III. PRELIMINARY ANALYSIS

Let  $t' \in (0, t_f]$  be a time instant at which the robot has reached a pick-up or drop-off location with zero velocity. The overall cost-to-go at state  $(q(t), \xi(t))$ ,  $t \in [0, t']$  is

$$J((q(t), \xi(t)), u|_{[t, t_f]}, p) = \ell(q(t), \xi(t), u|_{[t, t']}, p) + J(q(t'), \xi(t'), u|_{[t', t_f]}, p), \quad (5)$$

i.e., the sum of the time  $\ell$  until the next pick-up or drop-off at time  $t'$ , and the remaining time until the final state is reached. Assuming that all objects have been detected prior to  $t'$ , the second term on the right hand side of (5) is the cost of the optimal sequence of pick-ups and drop-offs, necessary for

completing the overall task. Let the set of all corresponding discrete state strings from the state  $q(t') = q$  to the final discrete state  $q_f$  be denoted by

$$\Sigma_q := \{\sigma = q_0 q_1 \dots q_d : (q_{i-1}, q_i) \in (\Pi \cup \Psi), i \in \{1, \dots, d\}, q_d = q_f, q_0 = q\}. \quad (6)$$

Minimizing the cost  $J_\sigma$  of a particular sequence  $\sigma \in \Sigma_q$  can be decoupled in terms of the input  $u|_{[t', t_f]}$  at every pick-up and drop-off time instant  $t_{i-1}, t_i \in [t', t_f]$ , i.e.,

$$\begin{aligned} J_\sigma^*(q(t'), \xi(t'), p) &= \min_{u|_{[t', t_f]}} J_\sigma(q(t'), \xi(t'), u|_{[t', t_f]}, p) \\ &= \sum_{i=1}^d \min_{u|_{[t_{i-1}, t_i]}} J(q_{i-1}, \xi(t_{i-1}), u|_{[t_{i-1}, t_i]}, p) \end{aligned} \quad (7)$$

with  $t_0 = t'$  and  $t_d = t_f$ . Assuming the absence of obstacles in  $\mathcal{Y}_g$ , the time-optimal motion of the robot with dynamics (2) from the hybrid state  $(q_{i-1}, (x(t_{i-1}) = [y(t_{i-1})^T \mathbf{0}_2^T]^T, \mathcal{Y}(t_{i-1})))$  to  $(q_i, (x(t_i) = [y(t_i)^T \mathbf{0}_2^T]^T, \mathcal{Y}(t_i)))$  with  $y(t_i), y(t_{i-1}) \in p \cup \{\mathbf{0}_2\}$  is on straight lines. Thus, using an affine transformation, (2) can be reduced to a double integrator in one dimensional space. The OCP for the reduced model corresponds to the classical linear time-OCP [2] solved by a piecewise constant control that takes values in the set  $\{\pm 1\}$  and yields the optimal cost  $t_i - t_{i-1} = 2\sqrt{m_{q_{i-1}} \|y(t_i) - y(t_{i-1})\|}$ . The controller can be transformed back to (2) by using the inverse affine transformation (details can be found in [8]). Since a transition from a hybrid state, where the robot with dynamics (??) has zero velocity, to another hybrid state, where the robot has zero velocity, can be tightly lower bounded by the cost for the time-optimal point-to-point motion of a double integrator with zero initial and final velocity, for the optimal cost of a string we obtain

$$J_\sigma^*(q(t'), \xi(t'), p) = \sum_{i=1}^d 2\sqrt{m_{q_{i-1}} \|y(t_i) - y(t_{i-1})\|}. \quad (8)$$

To illustrate this expression, consider a scenario with two remaining objects, both to be picked up and dropped. The corresponding discrete dynamics of  $\mathcal{H}$  are captured by the quadruple  $(Q, E, q_0, q_f)$  (Fig. 1), where  $Q$  and  $E$  are the corresponding sets of  $\mathcal{H}$ , and  $q_0$  and  $q_f$  are the initial and final discrete state (specified by Init and Fin), respectively. If both objects have been detected and the robot is at rest, the right hand side of (9) denotes the actual cost-to-go for completing the task. In addition, the right hand side of (9) can be used as a lower bound for the cost-to-go at time  $t'$ , where the robot is at rest but not both objects have been discovered, i.e.,

$$\begin{aligned} J((q(t'), \xi(t')), u|_{[t', t_f]}, p) &\geq \\ J_{lb}(q(t'), \xi(t'), p) &= \min_{\sigma \in \Sigma_q} J_\sigma^*(q(t'), \xi(t'), p), \end{aligned} \quad (9)$$

which represents the cost-to-go without taking into account exploration.

To obtain a finite conservative approximation for  $\mathcal{Y}(t)$ , introduce a finite cover of  $\mathcal{Y}_g$  by cells  $\omega_k, k \in \{1, \dots, K\}$

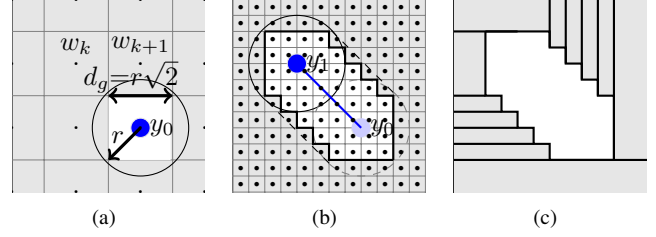


Fig. 2. A robot with sensing radius  $r$  over the coarsest allowed grid (a). A snapshot of the robot that has moved from  $y_0 = \mathbf{0}_2$  to  $y_1 \neq \mathbf{0}_2$  with  $d_g < r\sqrt{2}$  (b). The area covered along the path is under-approximated over the grid. The over-approximation  $\tilde{\mathcal{Y}}(t)$  (in gray) of  $\mathcal{Y}(t)$  is described by a finite number of rectangular regions (c) used for the time-driven optimization.

defined by a set of grid points  $W = \{w_1, \dots, w_K\}$ , equally spaced by  $d_g \leq r\sqrt{2}$ , such that  $\omega_k = \{y \in \mathcal{Y}_g : \|y - w_k\|_\infty \leq d_g/2\}$  (see Fig. 2 for an example). Let  $\mathcal{W}(t)$  denote the set of grid points, whose associated cells have not been completely covered by the robot's sensing range (1) until time  $t$ , i.e.  $\mathcal{W}(t) = \{w_i \in W : w_i \not\subseteq \bigcup_{\tilde{t} \in [0, t]} \mathcal{O}(y(\tilde{t}))\}$ . Thus,  $\mathcal{Y}(t)$  is over-approximated by  $\tilde{\mathcal{Y}}(t) = \{\bigcup_i w_i : w_i \in \mathcal{W}(t)\}$ . With that, we can turn to approximate solutions of Problems A and B.

#### IV. TIME-DRIVEN OPTIMIZATION

In this section, we present an approximation of Problem 1 based on equidistant time discretization.

##### A. Worst-case solution

By applying the min-max inequality and (9), since  $\ell \geq 0$ ,  $J_{lb} \geq 0$ , for the (certainty equivalent) worst-case evaluation of (5), we obtain

$$\begin{aligned} t_f^w - t &= \min_{u|_{[t, t_f]}} \max_p J((q(t), \xi(t)), u|_{[t, t_f]}, p) \\ &\geq \max_p ((\min_{u|_{[t, t']}} \ell(q(t), \xi(t), u|_{[t, t']}, p)) + J_{lb}(q(t'), \xi(t'), p)) \\ &\geq \min_{u|_{[t, t']}} \ell(q(t), \xi(t), u|_{[t, t']}, p^*) + J_{lb}(q(t') \xi(t'), p^*), \end{aligned} \quad (10)$$

where  $p^*$  is defined as

$$p^* = \arg \max_{p \in \tilde{\mathcal{Y}}(t)} J_{lb}(q(t') \xi(t'), p), \quad (11)$$

which follows from relaxing the assumption at time  $t'$  in a sense that the robot has zero velocity at pick-up or drop-off locations, but not all objects have necessarily been detected before  $t'$ . Since  $|\Sigma_q|$  is finite, it is possible to reformulate (11) by introducing a dummy variable  $\bar{t} \geq 0$  and  $|\Sigma_q|$  additional nonlinear constraints for each string in  $\Sigma_q$  leading to

$$\max_{\bar{t}, p \in \tilde{\mathcal{Y}}(t)} \bar{t}, \text{ s.t. } \forall \sigma \in \Sigma_q, J_\sigma^*(q(t'), \xi(t'), p) - \bar{t} \geq 0.$$

As the robot has zero velocity at  $t_0 = 0$ , the initial (approximately) optimal control  $u|_{[0, t_f^w]}$  can be obtained by solving (11) followed by re-translation to (2), as described in the previous section. Once the robot starts moving, optimizing the first

term of the third line of (10) at the optimum  $p^*$  is difficult in continuous time. Therefore, consider a finite equidistant sampling of a time horizon beginning at  $t$  with sampling time  $t_s$ , which we assume to include the yet unknown time  $t'$ , i.e.  $i \in [0, N_{\max} - 1]$ ,  $N_{\max} \in \mathbb{N}$ ,  $t = 0, t' < (N_{\max} - 1)t_s$ . Then, at every time instant, given the solution of (11), we solve

$$\begin{aligned} & \min_{u|_{[0, i-1]}} it_s, \\ & \text{s.t. (11), } \forall i \in [0, N_{\max} - 1], (q_i, \xi_i) \models \mathcal{C}(i), \end{aligned} \quad (12)$$

where  $\mathcal{C}(i)$  is the constraint set resulting from the corresponding discrete-time version of the hybrid automaton  $\mathcal{H}$ . The OCP can be approximately implemented as a MILP. For further implementation details, we refer the reader to Appendix A and [31] for a closely related OCP.

### B. Probabilistic solution

With (4), (5) and (9), prior to the discovery of all objects, the optimal cost in the probabilistic case is given by the minimum expected time (omitting function arguments)

$$\begin{aligned} E\{t_f - t\} & \geq \min_{u|_{[t, t_f]}} (E\{\ell\} + E\{J_{lb}\}) \\ & \geq \underbrace{\frac{1}{\kappa} \min_{u|_{[t, t']}} \int_{\tilde{\mathcal{Y}}(t)} \ell dp}_{\geq \min_{u|_{[t, t']}, p \in \tilde{\mathcal{Y}}(t)} \ell} + \underbrace{\frac{1}{\kappa} \int_{\tilde{\mathcal{Y}}(t)} \min_{\sigma \in \Sigma_q} J_{\sigma}^* dp}_{\geq \frac{1}{\kappa} \min_{\sigma \in \Sigma_q} \int_{\tilde{\mathcal{Y}}(t)} J_{\sigma}^* dp} \end{aligned} \quad (13)$$

where  $\kappa$  denotes the area of  $\tilde{\mathcal{Y}}(t)$ . The approximation of the first term follows from the fact that  $E\{\ell\}$  is certainly greater or equal to the shortest time needed for the robot to move from its current position to a currently unexplored point in  $\tilde{\mathcal{Y}}(t)$ , while the approximation of the second term is obtained by applying Jensen's inequality. To compute the control  $u|_{[t, t']}$  that minimizes the first term in (13), we formulate a MILP analogously to (12). The second term in (13) is obtained through numerical integration of  $J_{\sigma}$  for  $p$  over  $\tilde{\mathcal{Y}}(t)$ , followed by choosing the sequence  $\sigma^* \in \Sigma_q$  that yields the minimal cost. This allows for a receding horizon scheme that minimizes the cost-to-go at each time instant until all objects are dropped off.

## V. EVENT-DRIVEN OPTIMIZATION

The approaches presented in the previous section require solving computationally expensive MIPs at each time instant online. Since the locations of the objects are the only source of uncertainty in the considered problem, the ultimate goal is a tractable and scalable, albeit suboptimal alternative that avoids time discretization and requires re-computation only upon a detection. The approach proposed in the following is based on restricting the motion of the robot to a pre-specified family of curves, whose shape is determined by a finite parameter vector, such that the cost-to-go can be evaluated efficiently. This allows for an event-driven scheme based on an iterative gradient-based optimization over the parameters of the curve only upon detection.

Let the robot's position be described by the parametric equation

$$y(t) = c(s(t), \theta) = \begin{bmatrix} c_1(s(t), \theta) \\ c_2(s(t), \theta) \end{bmatrix} \in \mathbb{R}^2, \quad (14)$$

where  $s(t)$  denotes the position of the robot along the curve  $c$ ,  $\theta \in \mathbb{R}^a$  is a parameter vector that controls the shape of  $c$ , and  $c$  is twice continuously differentiable with respect to  $s$  and  $\theta$ . Let  $\tilde{s}(t)$  be the monotonically non-decreasing curve length function of  $c$  over  $t \in [0, t_f]$ . With  $\alpha \in \mathbb{R}_{>0}$  denoting the arc-length of  $c$ , let  $s(t) = \frac{\tilde{s}(t)}{\alpha}$  denote the normed arc-length variable, such that  $s(0)=0$  at the initial position, and  $s(t_f) = 1$  at the final position. The parametric functions employed in this work are Fourier series (see Appendix B) that exhibit rich expressiveness in terms of motion behaviors and allow for an efficient solution of the optimization problem. Other types of parametric functions or more complex robot dynamics may also be used, as outlined in Remark 2.

Upon detection, optimization will be performed by a bi-level optimization algorithm, based on iteratively solving the following two OCPs:

- 1) Find the parameter  $\theta^*$  that determines the optimal shape of  $c(s(t), \theta)$  solving Problem 1;
- 2) Control the motion of the robot along  $c(s(t), \theta^*)$  by the optimal  $s^*(t)$  that respects the restrictions imposed by  $\mathcal{H}$ .

The outline of the algorithm reads as follows: starting with an initial parameter guess for  $\theta$ , we solve the low-level OCP 2). Then,  $\theta$  is updated by solving 1) using the solution of 2). The high-level OCP is solved by an augmented Lagrangian method that allows for replacing the constrained optimization problem by a series of unconstrained optimization problems. Employing Infinitesimal Perturbation Analysis [33], we obtain the derivative of the augmented cost and solve the unconstrained OCPs by gradient-based methods. The steps 1) and 2) are solved iteratively until reaching a (local) minimum of the OCP, which is attained upon satisfying an iteration threshold condition. We start with solving the second step.

### A. Optimal motion along the curve

Let the first and second derivatives of (14) w.r.t.  $s$  be  $c'(s, \theta) = \partial c / \partial s$  and  $c''(s, \theta) = \partial^2 c / \partial s^2$ , respectively. Further, let  $\dot{s} = ds/dt$  and  $\ddot{s} = d^2s/dt^2$  denote the time derivatives. For the velocity and the acceleration along (14), we respectively obtain

$$\begin{aligned} \dot{c}(s, \theta) &= \frac{dc(s(t), \theta)}{dt} = c'(s, \theta) \dot{s}, \\ \ddot{c}(s, \theta) &= \frac{d^2c(s(t), \theta)}{dt^2} = c''(s, \theta) \dot{s}^2 + c'(s, \theta) \ddot{s}, \end{aligned}$$

and the robot's dynamics (2) are restated as

$$m_{q(t)}(c_i''(s, \theta) \dot{s}^2 + c_i'(s, \theta) \ddot{s}) = u_i, i \in \{1, 2\}. \quad (15)$$

With the employed arc-length parameterization, the robot traverses the curve at constant speed, i.e.,  $\|c'(s, \theta)\| = \alpha$ , where  $\alpha$  is the arc-length of  $c$ . Substituting  $u = u_c(t)[\cos(\varphi), \sin(\varphi)]^T$  in polar coordinates in (15) and using

$\|\ddot{c}(s, \theta)\| = \|u\|/m_{q(t)}$ , (2) is equivalently restated by (14) and the state  $\bar{x} = [s \ \dot{s}]^T$  with dynamics

$$\begin{aligned} \dot{\bar{x}}(t) &= \bar{f}_q(\bar{x}(t), u_c(t), \theta) \\ &= \begin{bmatrix} \bar{x}_2 & \sqrt{\frac{u_c^2(t)}{\alpha^2 m_{q(t)}^2} - \frac{(c'_1 c'_2 - c'_1 c''_2)^2 \bar{x}_2^4}{\alpha^4} - \frac{(c''_1 c'_1 + c''_2 c'_2) \bar{x}_2^2}{\alpha^2}} \end{bmatrix}^T. \end{aligned} \quad (16)$$

To simplify the analysis in the following, the necessary optimality conditions for  $u_c(t)$  will be derived for

$$\dot{\bar{x}}(t) \approx \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \bar{x}(t) + \frac{1}{\alpha m_{q(t)}} \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_c(t), \quad (17)$$

which represents a reasonable approximation of (16) along general Fourier series curves. Note that, for lines,  $c'_1 = c''_1 = 0$  implies  $c'_1 c'_2 - c'_1 c''_2 = 0$  and  $c''_1 c'_1 + c'_2 c''_2 = 0$ , and (17) describes the dynamics of the robot exactly. Since the sensor footprint (1) is typically much smaller than  $\mathcal{Y}_g$ , for evaluating the cost-to-go we assume that prior to their discovery all objects are located on (14), i.e.,  $\forall p^{(l)} \in p, p^{(l)} = c(s_l, \theta), s_l \in [0, 1]$ , and neglect the sensing range of the robot. A preliminary version of this analysis was presented in [32]. In what follows, we further assume that the high-level OCP (presented in the following section) provides an optimal parameter  $\theta^*$ , such that the robot moving along (14) with  $\theta^*$  plans to cover the remaining space  $\mathcal{Y}(t)$  as long as there are objects to be detected, and passes through object locations that have been discovered previously but have not been picked-up yet. We start the analysis assuming that there is only one object, i.e.  $O = \{o\}$ , with mass  $m$  located at  $\bar{x}_1 \in [0, 1]$ .

1) *Optimal control for one object:* The robot with dynamics (17) starts at  $\bar{x}(0) = [0 \ 0]^T$ ,  $q(0) = (\emptyset, \emptyset, \emptyset)$  and  $m_{q(0)} = m_\emptyset$ . Clearly, the optimal control solving Problem 1 is divided into three parts, i.e.  $u_c = (u_c^1, u_c^2, u_c^3)$ , denoting the control until detection, the control until pick-up and the control until drop-off. After the object is detected at time  $t_1$ , when the robot moves with velocity  $\bar{x}_2(t_1) \geq 0$ , it can be reached at time  $t_2$  with  $\bar{x}_2(t_2) = 0$  by employing a time-optimal bang-bang controller with a switching at time  $\tilde{t}_1$  [2], i.e.,

$$u_c^2(t) = \begin{cases} -1, & t \in [t_1, \tilde{t}_1), \\ 1, & t \in [\tilde{t}_1, t_2). \end{cases}$$

Solving (17) with  $u_c^2$  and  $m_{q(0)} = m_\emptyset$ , and applying the boundary conditions for the object's pick-up  $\bar{x}_1(t_1) = \bar{x}_1(t_2), \bar{x}_2(t_2) = 0$ , yields the optimal cost

$$(t_2 - t_1) = \underbrace{(1 + \sqrt{2})}_{c_g} \alpha m_\emptyset \bar{x}_2(t_1). \quad (18)$$

Since the robot stops at  $t_2$ , steering it back to the depot by  $u_c^3$  is again given by bang-bang control [2]. Since its corresponding cost is independent of  $\bar{x}(t)$ ,  $t \in [0, t_2]$ , it can be neglected for finding  $u_c^1$ .

In the worst case, the object is located the furthest away from the initial point, i.e., at  $s = 1$ . Thus, the time-optimal control satisfies the condition

$$u_{c,w}^{1*}(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x}_1 \in [0, 0.5), \\ -1, & \text{if } \bar{x}_1 \in [0.5, 1]. \end{cases} \quad (19)$$

In the probabilistic case, the object's location is uniformly distributed over  $[0, 1]$ . To compute  $u_{c,p}^{1*}$  we need to consider the time from detection to pick-up (18), yielding  $t_2 = t_1 + c_g \alpha m_\emptyset \bar{x}_2$ . To obtain a standard representation for the cost, introduce an additional state for the unknown detection time  $\tilde{x}_3 = t_1$ , leading to an extended system state  $\tilde{x} = [\bar{x}^T \ t_1]^T$  with dynamics (17) and  $\dot{\tilde{x}}_3 = 1$ . Substituting the relation  $d\tilde{x}_1 = \tilde{x}_2 dt$ , the expected time for picking up the object is

$$\begin{aligned} E\{t_2\} &= E\{\tilde{x}_3 + c_g \alpha m_\emptyset \tilde{x}_2\} \\ &= \int_0^1 (\tilde{x}_3 + c_g \alpha m_\emptyset \tilde{x}_2) d\tilde{x}_1 \\ &= \int_0^{t_2} (\tilde{x}_2 \tilde{x}_3 + c_g \alpha m_\emptyset \tilde{x}_2^2) dt, \end{aligned}$$

where  $t_2$  is free and the boundary constraints

$$\tilde{x}(0) = [0 \ 0 \ 0]^T, \quad \tilde{x}_1(t_2) = 1, \quad \tilde{x}_2(t_2) = 0$$

must be satisfied. Thus, the probabilistic OCP has been transformed into a free final time nonlinear OCP. The corresponding control Hamiltonian is

$$H(\tilde{x}, \lambda, u_{p,1}) = (\tilde{x}_2 \tilde{x}_3 + c_g \alpha m_\emptyset \tilde{x}_2^2) + \lambda^T(t) \begin{bmatrix} \tilde{x}_2 & \frac{u_{c,p}^1}{m_\emptyset \alpha} & 1 \end{bmatrix}^T, \quad (20)$$

with absolutely continuous costate dynamics

$$\dot{\lambda}(t) = - \frac{\partial H}{\partial \tilde{x}} = - \begin{bmatrix} 0 & 2c_g \alpha m_\emptyset \tilde{x}_2 + \tilde{x}_3 + \lambda_1 & \tilde{x}_2 \end{bmatrix}^T. \quad (21)$$

Applying Pontryagin's Minimum Principle, there exists an optimal state  $\tilde{x}^*$ , a control  $u_{c,p}^{1*}$ , and a nontrivial costate  $\lambda^*$  trajectory, such that  $\forall t \in [0, t_2]$ ,

$$H(\tilde{x}^*, \lambda^*, u_{c,p}^{1*}) \leq H(\tilde{x}^*(t), \lambda(t)^*, u_{c,p}^1(t)), \quad (22)$$

leading to the following theorems.

**Theorem 1.** *The optimal control is  $u_{c,p}^{1*}(t) \in \left\{-1, -\frac{1}{2c_g}, 1\right\}$  for almost all  $t \in [0, t_2]$ .*

**Theorem 2.** *The optimal control for  $\tilde{x}_1 \in [0, 1]$  is*

$$u_{c,p}^{1*}(\tilde{x}) = \begin{cases} 1, & \text{if } \tilde{x}_1 \in [0, \frac{1}{3+2\sqrt{2}}), \\ -\frac{1}{2+2\sqrt{2}}, & \text{if } \tilde{x}_1 \in [\frac{1}{3+2\sqrt{2}}, 1]. \end{cases} \quad (23)$$

The proofs can be found in Appendix C.

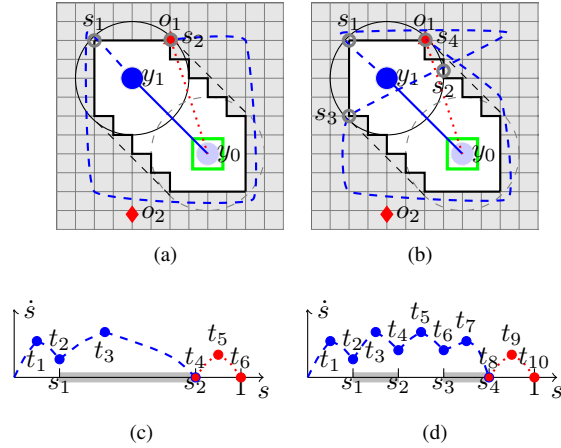


Fig. 3. Scenario upon detecting object  $o_1$ , denoted by  $\blacktriangle$ , and object  $o_2$ , denoted by  $\blacklozenge$ , with yet unknown position. Two planned position trajectories for the robot that has previously moved from  $y_0 = O_2$  to  $y_1 \neq O_2$  are shown with dashed lines in (a) and (b). The corresponding generalized trajectory  $\bar{x}|_{[t,t_f]}$  is shown in (c) and (d), respectively.

2) *Control for multiple objects*: For multi-object setups, we propose an approach to obtain the control analogously to the single-object case. While the worst-case optimal control is built from a finite sequence of bang-bang control segments, a computationally tractable scheme for the probabilistic control is derived as follows. Recall the robot moving in the position space shown in Fig. 2 (b) and consider a scenario with two objects. Like in the single-object case, the robot starts moving in the discrete state  $q = (\emptyset, \emptyset, \emptyset)$ . The possible discrete event strings until the robot comes to a halt for the first time consist of detecting an object followed by its immediate pick-up, i.e.  $\delta_i \pi_i$ , or detecting both objects and stopping at one of the two objects' positions, i.e.  $\delta_i \delta_j \pi_k$ ,  $i, k, j \in \{1, 2\}, i \neq j$  (see Fig. 1). For both cases, we employ the control (23) for the time interval up to the detection of the first object, although for more than one object this policy may not be optimal. Now assume that object  $o_1$  has just been detected. Fig. 3(a) and 3(b) show two possible curves that provide complete exploration of  $\tilde{\mathcal{Y}}(t)$ , allow for picking up  $o_1$  and end at the depot. The trajectories  $\bar{x}^*$  resulting from the probabilistic control are shown in Fig. 3(c) and 3(d), respectively. The particular control can be derived using the corresponding boundary and continuity conditions, as shown in Appendix D. The analysis for these two cases can be easily generalized to obtain the probabilistic control for an arbitrary finite number of unexplored segments in the interval  $[0, 1]$ .

Moving on to the problem with more than two objects, we obtain the controls along the curve by following a similar line of argumentation – both the worst-case and the probabilistic controls consist of a finite sequence of appropriate bang-bang control segments. For that, consider the set of discrete state strings  $\Sigma_q$  from a state  $q(t) = q$ , where all objects have been

detected, to the final discrete state  $q_f$ , as defined in (6). Then, let  $\Sigma_q|_{q_3 \setminus (q_1 \cup q_2)}$  denote the projection of all  $\sigma \in \Sigma_q$  onto  $q_3 \setminus (q_1 \cup q_2)$ . For a given  $\theta$  and with  $t = t_0$ , the curve (14) yields a discrete state string  $\sigma \in \Sigma_q|_{q_3 \setminus (q_1 \cup q_2)}$  that is traversed in time

$$J_1 = \sum_{i=1}^d (t_i - t_{i-1}) = \sum_{i=1}^d \int_{\bar{x}_1(t_{i-1})}^{\bar{x}_1(t_i)} \frac{1}{\bar{x}_2} d\bar{x}_1. \quad (24)$$

Since minimizing (24) over the free parameters of the proposed policy, i.e. the switching times, can be decoupled at pick-up and drop-off instants, we solve  $d - 1$  Two-Point Boundary Value Problems (TPBVP's)

$$\begin{aligned} \bar{x}^*|_{[t_{i-1}, t_i]} = \arg \min_{\bar{x}_1|_{[t_{i-1}, t_i]}} \int_{\bar{x}_1(t_{i-1})}^{\bar{x}_1(t_i)} \frac{1}{\bar{x}_2} d\bar{x}_1, \\ \text{s.t. (16), } \bar{x}_2(t_{i-1}) = \bar{x}_2(t_i) = 0, \end{aligned} \quad (25)$$

using the corresponding control trajectory  $u_c$  for the worst-case or the probabilistic problem. A segment  $\bar{x}^*|_{[t_{i-1}, t_i]}$  of  $\bar{x}^* = (\bar{x}^*|_{[t_0, t_1]}, \dots, \bar{x}^*|_{[t_{d-1}, t_d]})$  is obtained by solving an initial value problem backward and forward in time, and choosing the minimal velocity  $\bar{x}_2$  along the curve. Clearly, once all objects are detected, both versions A) and B) of Problem 1 become deterministic and can be easily solved.

**Remark 2.** The TPBVP (25) can be easily extended for general convex input constraints  $u \in U$  or nonlinear robot dynamics, as shown in [36]. If the curve is not continuously differentiable, it can be approximated by a finite number of cubic spline segments, such that the OCPs remain numerically feasible.

Now the only remaining task is computing  $\theta^*$  that solves the high-level optimization problem.

### B. Parametric optimization

The solution  $\theta^*$  of the high-level optimization problem yields a curve (14) that can be traversed optimally according to the policy outlined above, while providing complete exploration of the remaining space  $\tilde{\mathcal{Y}}(t)$ , passing through all positions of objects in  $q_3 \setminus (q_1 \cup q_2)$  and ending at the depot. Let these constraints be captured by the mapping  $h : \bar{x}|_{[t, t_f]} \times \mathbb{R}^a \rightarrow \mathbb{R}^b$ , where  $h(\bar{x}^*, \theta) = \mathbf{0}_b$ , if the constraints are satisfied, and  $h(\bar{x}^*, \theta) \neq \mathbf{0}_b$ , else. Thus, with (24) and (25), the high-level optimization problem at  $(q(t), (\bar{x}(t), \mathcal{W}(t)))$  reads

$$\begin{aligned} (t_f - t)^* = \min_{\theta} J_1 \\ \text{s.t. } h(\bar{x}^*, \theta) = \mathbf{0}_b. \end{aligned} \quad (26)$$

Leveraging ideas from [37], [30], let  $J_2 : \bar{x}|_{[t, t_f]} \times \mathbb{R}^a \rightarrow \mathbb{R}^b$  be a continuously differentiable version of the mapping  $h$  that captures the constraints of the high-level OCP, such that  $J_2(\bar{x}^*, \theta) = \mathbf{0}_b$ , if  $h(\bar{x}^*, \theta) = \mathbf{0}_b$ , and  $J_2(\bar{x}^*, \theta) > \mathbf{0}_b$ , else. We will now introduce the individual constraints of the optimization problem.

To guarantee that all objects are detected eventually, the robot has to observe every  $w_{i_k} \in \mathcal{W}(t) \subset W, i_k \in$



$\{1, \dots, K\}$ , when it moves along the curve. For every  $w_{i_k}$ , we define a function  $d_k(s, \theta)$ , such that  $d_k(s, \theta) = 0$ , if  $w_k$  is within the sensing range (1) of the robot with position  $y(t) = c(s, \theta)$ , and  $d_k(s, \theta) > 0$ , otherwise. The choice of  $d_k(s, \theta)$  is not unique. In particular, we employ

$$d_k(s, \theta) = \begin{cases} 1 - \exp(-(D_k(s, \theta) - r)^2), & \text{if } D_k(s, \theta) > r, \\ 0, & \text{else,} \end{cases} \quad (27)$$

where  $D_k(s, \theta) = \|c(s, \theta) - w_k\|$ . This leads to the constraint vector  $\mathcal{D}_{\mathcal{W}} = [\min_{s \in [0, 1]} d_1(s, \theta), \dots, \min_{s \in [0, 1]} d_{\tilde{K}}(s, \theta)]$ ,  $\tilde{K} = |\mathcal{W}(t)|$ , which is required to be equal to  $\mathbf{0}_{1, \tilde{K}}$ , as every  $w_{i_k} \in \mathcal{W}(t)$  has to be seen along the curve.

In addition, all objects that have been detected, but have not been picked up or dropped at the depot yet, i.e. all  $o_l \in q_3 \setminus (q_1 \cup q_2)$ , have to be eventually picked up, while the robot moves along the curve. Since the requirement that the robot performs a pick-up with zero velocity is taken care of by the low-level OCP, in the high-level OCP we only have to guarantee that the curve passes through all corresponding positions  $p^{(i)}$ . Hence, for every  $p^{(i)}, i_l \in \{1, \dots, \tilde{L}\}$ ,  $\tilde{L} = |q_3 \setminus (q_1 \cup q_2)|$ , we define a function  $\tilde{d}_l(s, \theta)$ , such that  $\tilde{d}_l(s, \theta) = 0$ , if  $p^{(i)} = c(s, \theta)$ , and  $\tilde{d}_l(s, \theta) > 0$ , otherwise. In particular, we employ  $\tilde{d}_l(s, \theta) = 1 - \exp(-\|c(s, \theta) - p^{(i)}\|^2)$ . Note that the choice of  $\tilde{d}_l$  is not unique. This leads to the constraint vector  $\mathcal{D}_{q_3 \setminus (q_1 \cup q_2)} = [\min_{s \in [0, 1]} \tilde{d}_1(s, \theta), \dots, \min_{s \in [0, 1]} \tilde{d}_{\tilde{L}}(s, \theta)]$ , which is required to be equal to  $\mathbf{0}_{1, \tilde{L}}$ , since every  $p^{(i)}$  has to be visited at some point along the curve.

Further, upon a detection at  $s_- = s(t)$ , if the robot's velocity is not zero, the robot is required to continue its motion in a smooth manner, despite changing from moving along the curve characterized by the previous parameter vector denoted by  $\theta_-$ , to moving along the curve characterized by the current parameter vector denoted by  $\theta$ . Hence, we introduce a function  $d_{in}^y(s_-, \theta_-, \theta)$ , such that  $d_{in}^y(s_-, \theta_-, \theta) = 0$  when  $c(s_-, \theta_-) = c(0, \theta)$ , and  $d_{in}^y(s_-, \theta_-, \theta) > 0$ , else. In particular, we employ  $d_{in}^y(s_-, \theta_-, \theta) = 1 - \exp(-\|c(s_-, \theta_-) - c(0, \theta)\|^2)$ . Analogously, introduce a function  $d_{in}^v(s_-, \theta_-, \theta)$ , such that  $d_{in}^v(s_-, \theta_-, \theta) = 0$  when  $c'(s_-, \theta_-) = c'(0, \theta)$ , and  $d_{in}^v(s_-, \theta_-, \theta) > 0$ , else. In particular, we employ  $d_{in}^v(s_-, \theta_-, \theta) = 1 - \exp(-\|c'(s_-, \theta_-) - c'(0, \theta)\|^2)$ . Note that both  $d_{in}^y(s_-, \theta_-, \theta)$  and  $d_{in}^v(s_-, \theta_-, \theta)$  are not unique. Finally, the robot is required to return to the depot in order to drop all objects. Since the requirement that the robot performs a drop-off with zero velocity is taken care of by the low-level OCP, in the high-level OCP we just have to guarantee that the curve ends at the depot. Thus, we introduce the function  $d(\theta)$ , such that  $d(\theta) = 0$ , if  $c(1, \theta) = y_d$ , and  $d(1, \theta) > 0$ , otherwise. In particular, we use  $d(\theta) = 1 - \exp(-\|c(1, \theta) - y_d\|^2)$ . Note that the choice of

$d(\theta)$  is not unique. This leads to the constraint vector

$$\mathcal{D}_{in, f} = \begin{cases} [d_{in}^y, d_{in}^v, d], & \text{if } \bar{x}_2 \neq 0, \\ [d_{in}^y, d], & \text{else,} \end{cases}$$

which is required to be equal to zero. Once all objects are discovered, exploration is no longer necessary and the constraints  $\mathcal{D}_{\mathcal{W}}$  are neglected. Thus, the constraints  $J_2$  are given by

$$J_2(\bar{x}^*, \theta) = \begin{cases} [\mathcal{D}_{\mathcal{W}}, \mathcal{D}_{q_3 \setminus (q_1 \cup q_2)}, \mathcal{D}_{in, f}]^T, & \text{if } q_3(t) \neq O, \\ [\mathcal{D}_{q_3 \setminus (q_1 \cup q_2)}, \mathcal{D}_{in, f}]^T, & \text{else.} \end{cases} \quad (28)$$

The constrained optimization problem (26) is approximately solved with (28) by the augmented Lagrangian method [38], yielding the iterative unconstrained optimization problem

$$\theta_z^* = \arg \min_{\theta} \hat{J} = J_1 + \frac{\mu_z}{2} J_2^T J_2 + \bar{\lambda}_z^T J_2, \quad (29)$$

where  $\mu_z \in \mathbb{R}_{>0}$  is an optimization tuning variable that increases with each iteration and  $\bar{\lambda}_z \in \mathbb{R}^b$  is an estimate of the Lagrangian multiplier, updated by

$$\bar{\lambda}_{z+1} = \bar{\lambda}_z + \mu_z J_2(\bar{x}^*, \theta_z^*) \quad (30)$$

for every iteration  $z = 0, 1, 2, \dots$

Since (28) is continuously differentiable w.r.t.  $s$  and  $\theta$ , the unconstrained optimization problem (29) can be solved by gradient-based optimization. Taking into account the dynamics of the hybrid automaton  $\mathcal{H}$  and substituting  $\alpha = \|c'(\bar{x}_1, \theta)\|$  for the employed arc-length parameterization, the gradient of (29) (omitting function arguments) is given by

$$\begin{aligned} \nabla_{\theta} \hat{J} &= \sum_{n=1}^N \nabla_{\theta} \int_{t_{n-1}}^{t_n} dt + \sum_{\beta=1}^b (\mu_z J_{2, \beta} + \bar{\lambda}_{z, \beta}) \nabla_{\theta} J_{2, \beta} \\ &= \sum_{n=1}^N \nabla_{\theta} \int_{t_{n-1}}^{t_n} \frac{\|c'\|}{\alpha} dt + \sum_{\beta=1}^b (\mu_z J_{2, \beta} + \bar{\lambda}_{z, \beta}) \nabla_{\theta} J_{2, \beta}. \end{aligned} \quad (31)$$

Observing that  $\bar{x}_1$  depends on  $\theta$  through (16), we employ Infinitesimal Perturbation Analysis (IPA) [33] to obtain the gradient  $\nabla_{\theta} \int_{t_{n-1}}^{t_n} \frac{\|c'\|}{\alpha} dt$  (and, thus,  $\nabla_{\theta} \hat{J}$ ). Over an interval  $\tau_n = [t_{n-1}, t_n]$ ,  $n = 1, \dots, N$ , the evolution of  $\bar{x}$  is described by the vector field  $\bar{f}_{n-1}(t, \bar{x}, \theta)$  (with a slight abuse of notation). For  $t \in [t_{n-1}, t_n]$ ,

$$\frac{d}{dt} \nabla_{\theta} \bar{x}(t, \theta) = \frac{\partial \bar{f}_{n-1}(t, \bar{x}, \theta)}{\partial \bar{x}} \nabla_{\theta} \bar{x}(t, \theta) + \nabla_{\theta} \bar{f}_{n-1}(t, \bar{x}, \theta) \quad (32)$$

holds with the boundary condition

$$\begin{aligned} \nabla_{\theta} \bar{x}(t_{n-1}^+, \theta) &= \nabla_{\theta} \bar{x}(t_{n-1}^-, \theta) + \\ &[\bar{f}_{n-2}(t_{n-1}^-, \bar{x}, \theta) - \bar{f}_{n-1}(t_{n-1}^+, \bar{x}, \theta)] \nabla_{\theta} t_{n-1}(\theta). \end{aligned} \quad (33)$$

Thus, we obtain

$$\nabla_{\theta} \bar{x}(t, \theta) = \nabla_{\theta} \bar{x}(t_{n-1}^+, \theta) + \int_{t_{n-1}}^t \frac{d}{dt} \nabla_{\theta} \bar{x}(t, \theta) dt. \quad (34)$$



Since  $\bar{f}_{1,n-2}(t_{n-1}^-, \bar{x}, \theta) = \bar{f}_{1,n-1}(t_{n-1}^+, \bar{x}, \theta)$ , from (33), we observe that  $\nabla_{\theta} \bar{x}_1(t_{n-1}^+, \theta) = \nabla_{\theta} \bar{x}_1(t_{n-1}^-, \theta)$ . Using (32) with  $\frac{\partial \bar{f}_{1,n-1}(t, \bar{x}, \theta)}{\partial \bar{x}_1} = 0$  and  $\nabla_{\theta} \bar{f}_{1,n-1}(t, \bar{x}, \theta) = 0$  and (34), the gradient (31) (omitting function arguments) is obtained by

$$\nabla_{\theta} \hat{J} = \sum_{i=1}^d \int_{t_{i-1}}^{t_i} \nabla_{\theta} \frac{\|c'(\bar{x}_1, \theta)\|}{\alpha} dt + \sum_{\beta=1}^b (\mu_z J_{2,\beta} + \bar{\lambda}_{z,\beta}) \nabla_{\theta} J_{2,\beta}. \quad (35)$$

The partial derivatives of (27) w.r.t.  $s$  or  $\theta$  (omitting function arguments) are obtained by

$$\nabla_{(\cdot)} d_k = \begin{cases} 2(D_k - r) \exp(-(D_k - r)^2) \frac{\partial D_k}{\partial (\cdot)}, & \text{if } D_k > r, \\ 0, & \text{else,} \end{cases} \quad (36)$$

where  $\frac{\partial D_k}{\partial (\cdot)} = \frac{1}{D_k} ((c_1 - w_{1,k}) \nabla_{(\cdot)} c_1 + (c_2 - w_{2,k}) \nabla_{(\cdot)} c_2)$ . The corresponding derivatives of  $\tilde{d}_l$  and  $d$  are obtained analogously. The required partial derivatives for Fourier series are shown in Appendix B. Then, to compute (35), we obtain the optimal solutions of the internal OCPs of the constraints  $\mathcal{D}_{\mathcal{W}}$  by a gradient-based algorithm, i.e.,

$$s_{z+1} = s_z - \eta_z \left. \frac{\partial d_k(s_z, \theta)}{\partial s} \right|_{s \in [0,1]}, \quad (37)$$

where  $\{\eta_z\}, z = 0, 1, 2, \dots$  is an appropriate step-size sequence,  $\left. \frac{\partial d_k}{\partial s} \right|_{s \in [0,1]}$  is the gradient projected onto the feasible interval  $s \in [0, 1]$  and the algorithm terminates when  $\left| \left. \frac{\partial d_k}{\partial s} \right|_{s \in [0,1]} \right| < \epsilon$  (for a given threshold  $\epsilon$ ). Analogously, we solve the internal OCPs of the constraints  $\mathcal{D}_{q_3 \setminus (q_1 \cup q_2)}$ , yielding the value of  $J_2$ . Then,  $\nabla_{\theta} J_2$  is computed by (36) for the (local) optima acquired by solving (37) for the corresponding constraints.

As the considered setup is static, the gradient obtained through IPA is a trivially unbiased estimate of the gradient of the cost for Problem B. Thus, the optimal parameter vector  $\theta^*$  is obtained with a gradient-based algorithm, i.e.,

$$\theta_{z+1} = \theta_z - \eta_z \nabla_{\theta} \hat{J}(\bar{x}^*, \theta_z) \Big|_{\mathcal{Y}_g} \quad (38)$$

where  $\{\eta_z\}, z = 0, 1, 2, \dots$  is a properly selected step-size sequence and the gradient is projected onto the feasible position space  $\mathcal{Y}_g$ . The algorithm terminates when  $|\nabla_{\theta} \hat{J}(\bar{x}^*, \theta_z)|_{\mathcal{Y}_g}| < \epsilon$  for a pre-specified threshold  $\epsilon$ . Note that the OCPs (37) are non-convex, the solution of (29) acquired by (38) will, in general, be only locally optimal.

The overall event-driven solution is summarized in Alg. 1. Upon a detection of an object, as long as the current solution violates the constraints, (25) and (38) are solved iteratively, where  $\mu_z$  increases with each iteration, thus increasing the importance of  $J_2$  over  $J_1$  in the optimization. A detection yields an additional constraint for the high-level OCP, which will be violated initially, in general. Assuming that all other constraints were satisfied in the previous run, the algorithm typically terminates within a small number of iterations.

---

**Algorithm 1** Event-driven receding horizon control upon detection

---

**Input:** Robot dynamics described by the hybrid automaton  $\mathcal{H}$  with  $\text{Init} = (q(t), (\bar{x}(t), \mathcal{W}(t)))$ ; curve  $y(t) = c(\bar{x}_1, \theta)$  with initial parameter vector  $\theta$ ; optimization parameters  $0 < \epsilon \ll 1, \nu > 1$

**Output:** The optimal control  $u|_{[t, t_f]}^*$

---

- 1: Set  $\mu_1 = 1$  and  $\bar{\lambda}_1 = \mathbf{0}_b$ .
  - 2: **while**  $J_2(\bar{x}^*, \theta) > \epsilon \mathbf{1}_b$  **do**
  - 3:   **repeat**
  - 4:     Compute  $u^*(\bar{x}^*)$  through (25) and  $J_2$  and  $\nabla_{\theta} J_2$  through (37) for  $\theta$  for the worst-case or the probabilistic case.
  - 5:     Compute  $\hat{J}$  and  $\nabla_{\theta} \hat{J}|_{\mathcal{Y}_g}$  with  $\mu_z$ , and update  $\theta$  through (38) and  $\bar{\lambda}_z$  through (30).
  - 6:   **until**  $|\nabla_{\theta} \hat{J}|_{\mathcal{Y}_g}| < \epsilon$
  - 7:   Set  $\mu_{z+1} = \nu \mu_z$ .
  - 8: **end while**
  - 9: **return**  $u|_{[t, t_f]}^* = u^*(\bar{x}^*)$
- 

Before the first detection, a good initial guess for  $\theta$  is required, such that Alg. 1 produces a good local optimum despite the non-convexity of (29). A good initial guess is often obtained by randomized optimization, e.g. using a Stochastic Comparison Algorithm [39].

**Remark 3.** Under the presence of an obstacle  $obs$  with a-priori unknown size and location, the obstacle is approximated by all  $w_k$  of the discretization of  $\mathcal{Y}_g$  that have been covered by the sensor until time  $t$  and belong to the obstacle region, i.e. all  $w_k \in \cup_{\tilde{t} \in [0, t]} \mathcal{O}(y(\tilde{t})) \wedge w_k \in obs$ . Then,  $J_2$  is augmented by a term that is only active in the surroundings  $d_s = d_g / \sqrt{2}$  of the corresponding  $w_k, k \in \{1, \dots, \tilde{K}\}$ , and is continuously differentiable w.r.t.  $s$  or  $\theta$ , e.g.,

$$J_o = \sum_{k=1}^{\tilde{K}} \max\{0, 1 - (D_k/d_s)^2\}^2. \quad (39)$$

With the partial derivative

$$\nabla_{(\cdot)} J_o = -\frac{4}{d_s^2} \sum_{k=1}^{\tilde{K}} \left[ \frac{\partial D_k}{\partial (\cdot)} \max\{0, 1 - \frac{D_k^2}{d_s^2}\} \right],$$

the (local) minimum of  $J_o^*$  is acquired by solving an OCP of the form (37), and the corresponding  $\nabla_{\theta} J_o$  is used to solve (38). Note that  $J_o = 0$  and  $\nabla_{\theta} J_o = \mathbf{0}_a$ , if the curve does not intersect the obstacle. Alg. 1 remains unchanged. In general, the non-convexity of the optimization space increases with a growing number of obstacles, which may deteriorate the quality of the local minima obtained by Alg. 1. In environments with high obstacle densities, one should choose parameterizations with a higher degree of freedom (i.e. larger  $\Gamma_1$  and  $\Gamma_2$  in the case of Fourier series) to allow for efficient obstacle avoidance.

### C. Complexity

The time-driven approaches are computationally expensive due to their NP-hardness and the generally increasing non-convexity of  $\hat{\mathcal{Y}}(t)$  during mission execution. The number of possible discrete state strings  $|\Sigma_q|$  is exponential in the number of not yet detected objects. In our implementation, the time-driven approaches remained computationally feasible for  $|L| \leq 4$ . In contrast, the event-driven approaches rely on solving a finite number of OCPs by gradient-based methods with significantly lower computational cost, as indicated by the numerical example in the following.

### VI. NUMERICAL EXAMPLE

1) *Implementation*: The methods were implemented in MATLAB, using the solver IPOPT for the MIPs of the time-driven and *ode45* for the event-driven methods. All computations were performed on an Intel® Core™ i7 2.20 GHz processor with 8 GB RAM.

2) *Setup*: Consider the bounded position space  $\mathcal{V}_g = [-5, 5] \times [-5, 5]$  m and its regular discretization (as described in Section III) with a grid constant of 0.25 m. Let the object set be  $O = \{o_1, o_2, o_3\}$  with  $p^{(1)} = [-3.1, -3.1]^T$ ,  $m^{(1)} = 1$  kg,  $p^{(2)} = [1.9, -1.9]^T$ ,  $m^{(2)} = 2$  kg and  $p^{(3)} = [3, 3]^T$ ,  $m^{(3)} = 2$  kg. Initially, the positions of the objects are unknown to the robot. The robot with sensor footprint of size  $r = 1$  m and nominal mass  $m_0 = 2$  kg starts at the depot  $y_d = \mathbf{0}_2$  at rest. A static obstacle with a priori unknown location and size to the robot, is described by

$$\text{obs} = \left\{ y \in \mathbb{R}^2: \begin{bmatrix} \mathbf{I}_2 \\ -\mathbf{I}_2 \end{bmatrix} y \leq \begin{bmatrix} \mathbf{I}_2 \\ -\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} 1.875 \\ -3.625 \end{bmatrix} + \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} 0.375 \\ 0.375 \end{bmatrix} \right\}.$$

The sampling time  $t_s = 0.2$  s and the optimization horizon  $N_{\max} = 8$  were chosen for the time-driven approaches. Fourier series of order  $\Gamma_1 = \Gamma_2 = 3$  were used for the event-driven approaches. Optimization was performed with parameters  $\nu = 2$ ,  $\epsilon = 10^{-3}$ , and a maximal integration time of 40 s for *ode45*. The parameter vector  $\theta$ , which characterizes the curve before the first detection, is obtained by running Alg. 1 for 100 random initializations and selecting the best (local) optimum.

3) *Analysis*: Fig. 4 shows snapshots of the robot's motion at object detection instants and the final time, obtained by applying the presented methods. Snapshots at obstacle detection instants were omitted due to space limitations. As it can be seen in the plots, the robot successfully avoids the obstacle. The time-driven and the event-driven methods lead to qualitatively similar solutions with small performance differences. On average, online re-computation took 35 s and 21 s for the worst-case and probabilistic time-driven approaches (performed at every time instant), and 6 s and 5 s for the worst-case and probabilistic event-driven approaches (performed only upon a detection), respectively, thus indicating that the latter two are particularly suitable for real-time computation.

For 100 random placements of the three objects in  $\mathcal{V}_g$ , the average cost for solving the task by the probabilistic event-driven solution was 30.1 s, thus, performing better than

the worst-case event-driven solution with an average cost of 32.21 s. In general, the worst-case solution results in a more “cautious” policy including intermediate pick-ups and drop-offs. In contrast, the probabilistic evaluation typically leads to a “threshold-based” policy, where previously detected objects are collected in one sweep after longer exploration phases. Note that both the convergence speed and the quality of the outcomes of the event-driven approaches strongly depend on the chosen initial conditions and the step size selection method of the gradient optimization procedures.

### VII. CONCLUSIONS

A time-optimal hybrid control problem for a robot that has to find and collect a finite number of objects located in a two-dimensional space and move them to a depot has been addressed. Two approaches have been proposed for the worst- and a probabilistic case, assuming uniform distribution of the objects over the search space – a time-driven scheme based on a space and time discretization, and an event-driven approach based on motion parameterization. The methods were compared in a numerical example, reflecting the significant computational advantage of the event-driven methods, while yielding similar qualitative results to the time-driven ones. The worst-case evaluation of the cost typically provides a “cautious” solution, often resulting in an intermediate pick-up and drop-off, while the probabilistic evaluation leads to a “threshold-based” policy, consisting of longer exploration phases and multiple successive pick-up's with a following drop-off. Future work will address a multi-robot setup and alternative optimization techniques for the event-driven approaches, e.g. using the Alternating Direction Method of Multipliers (ADMM) [40].

#### APPENDIX A: CONSTRAINT SET

The constraint set  $\mathcal{C}(i)$  at time  $i$  is constructed successively for  $\mathcal{H}$  by introducing additional slack variables. We provide a brief overview of the most important aspects of the implementation based on examples. Let  $i, j \in [0, N_{\max} - 1]$  be time instants in the optimization horizon. A depot visit at  $i$  is captured by a Boolean vector  $b_{d_i}$  with

$$b_{d_i} = \begin{cases} 1, & \text{if } x_i = [y_d^T \ \mathbf{0}_2^T]^T \wedge b_{d_j} = 0, \forall j < i, \\ 0, & \text{else.} \end{cases}$$

Then, a switching of the time-discretized dynamics (obtained by equidistant sampling of (2)) from  $q_1$  to  $q_0$  is realized by

$$\forall i, x_{i+1} = \begin{cases} \tilde{f}_{q_1}(x_i, u_i), & \text{if } b_{d_j} = 0, \forall j \leq i, \\ \tilde{f}_{q_0}(x_i, u_i), & \text{else,} \end{cases}$$

implying that the system evolves in state  $q_1$  until  $b_{d_j} = 1$ , when it switches to and remains in  $q_0$  thereafter. The input constraints are approximated by a regular  $Z$ -sided polygon, i.e.,

$$\forall \epsilon \in \{1, \dots, Z\}, \forall i, u_{i,1} \sin\left(\frac{2\pi\epsilon}{Z}\right) + u_{i,2} \cos\left(\frac{2\pi\epsilon}{Z}\right) \leq 1.$$

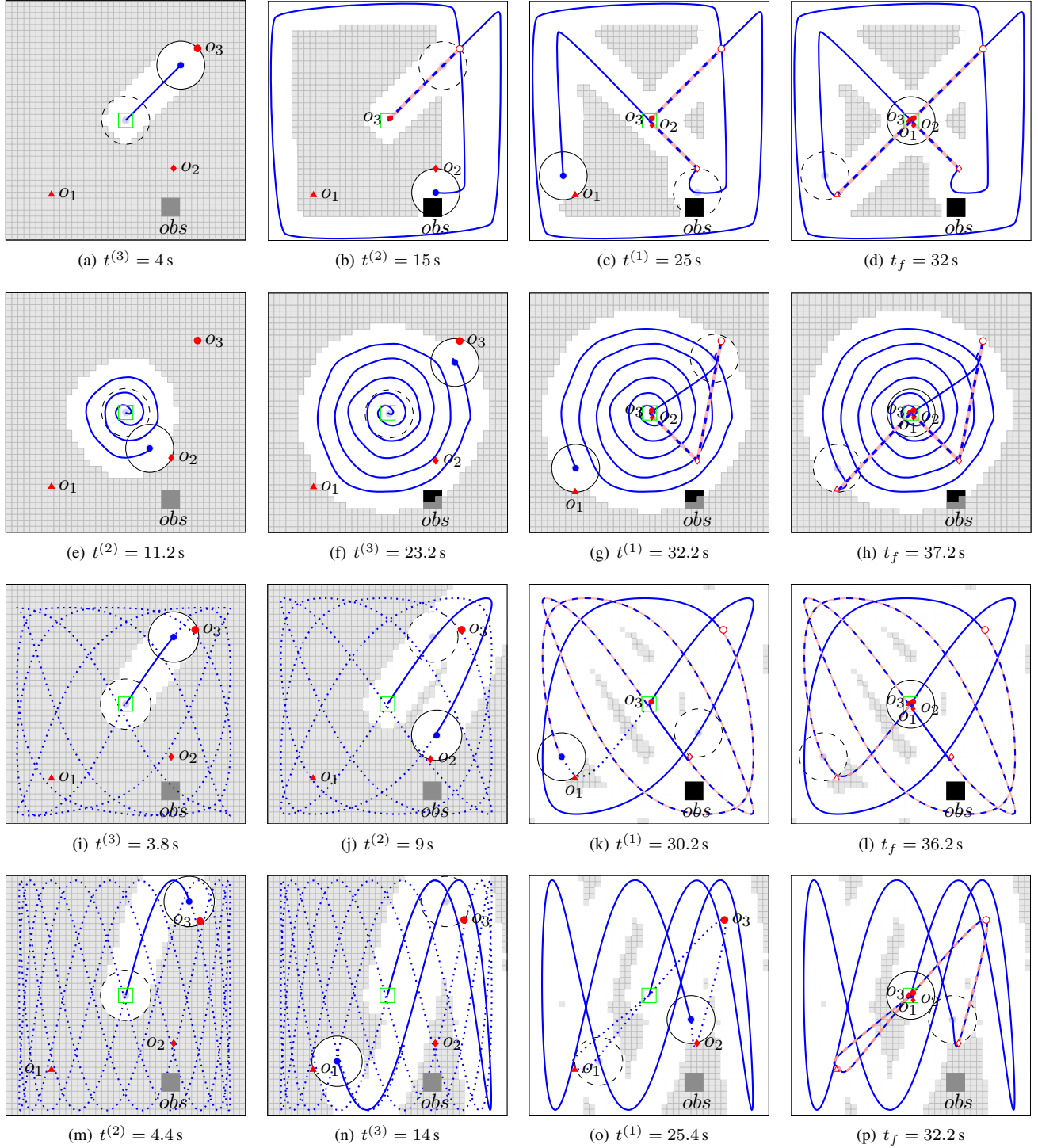


Fig. 4. Snapshots of the robot's motion at detection instants  $t^{(l)}$  and the final time  $t_f$  respectively obtained with the time-driven worst-case (a-d) and probabilistic (e-h), and the event-driven worst-case (i-l) and probabilistic (m-p) methods. Executed paths with nominal dynamics are denoted by solid, executed paths in other dynamical modes by thicker dashed, and planned trajectories by dotted lines.

The final time constraint can be captured by a boolean vector

$$\forall i, b_{f_i} = \begin{cases} 1, & \text{if } x_i = [y^T \mathbf{0}_2^T]^T \wedge y \in \{p^*, y_d\}, \\ 0, & \text{else,} \end{cases}, \quad \sum_{i=0}^{N_{\max}-1} b_{f_i} = 1,$$

reflecting that the trajectory may end at an expected position  $p^*$  of an object or at the depot. The constraints are implemented by big- $M$  relaxations, commonly employed in MIP, where the relaxation coefficients are chosen according to the setup. Finally, the cost is  $\ell = t_s \sum_{i=1}^{N_{\max}-1} i b_{f_i}$ .

## APPENDIX B: FOURIER SERIES TRAJECTORIES

In the event-driven approaches, we parameterize the position of the robot by Fourier series of respective order  $\Gamma_1$  and  $\Gamma_2$ , i.e.,

$$y(t) = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} a_0^1 + \sum_{\gamma=1}^{\Gamma_1} a_\gamma^1 \sin(4\pi^2 \gamma f_1 s(t) + \phi_\gamma^1) \\ a_0^2 + \sum_{\gamma=1}^{\Gamma_2} a_\gamma^2 \sin(4\pi^2 \gamma f_2 s(t) + \phi_\gamma^2) \end{bmatrix},$$

where  $f_1$  and  $f_2$  are base frequencies,  $a_0^1$  and  $a_0^2$  are zero frequency components,  $a_\gamma^1$  and  $a_\gamma^2$  are amplitudes for the sinusoid functions with frequency  $\gamma f_1$  and  $\gamma f_2$ , and  $\phi_\gamma^1$  and  $\phi_\gamma^2$  are phase differences with respect to the  $(\gamma + 1)$ -th term of  $y_1$  or  $y_2$ . Since only the ratio of  $f_1$  and  $f_2$  (and not their absolute values) determines the shape of the trajectories,  $f_1$  is treated as a free parameter, while  $f_2$  is kept constant. With  $A_1 = [a_0^1, \dots, a_{\Gamma_1}^1]$ ,  $A_2 = [a_0^2, \dots, a_{\Gamma_2}^2]$ ,  $\Phi_1 = [\phi_1^1, \dots, \phi_{\Gamma_1}^1]$  and  $\Phi_2 = [\phi_1^2, \dots, \phi_{\Gamma_2}^2]$ , the overall parameter vector is  $\theta = [f_1, A_1, A_2, \Phi_1, \Phi_2]^T$ . The derivative of the curve w.r.t.  $\theta$  with  $i \in \{1, 2\}$  is

$$\nabla_\theta y_i = \nabla_\theta c_i(s, \theta) = \begin{bmatrix} \frac{\partial y_i}{\partial f_1} & \frac{\partial y_i}{\partial A_1} & \frac{\partial y_i}{\partial A_2} & \frac{\partial y_i}{\partial \Phi_1} & \frac{\partial y_i}{\partial \Phi_2} \end{bmatrix}^T$$

with  $\gamma = 1, \dots, \Gamma_i$  and

$$\begin{aligned} \frac{\partial y_i}{\partial f_1} &= \begin{cases} 4\pi^2 s(t) \sum_{\gamma=1}^{\Gamma_1} a_\gamma^1 \cos(4\pi^2 \gamma f_1 s(t) + \phi_\gamma^1), & i=1, \\ 0, & i=2, \end{cases} \\ \frac{\partial y_i}{\partial a_0^j} &= \begin{cases} 1, & i=j, \\ 0, & i \neq j, \end{cases} \quad \frac{\partial y_i}{\partial a_\gamma^j} = \begin{cases} \sin(4\pi^2 \gamma f_j s(t) + \phi_\gamma^j), & i=j, \\ 0, & i \neq j, \end{cases} \\ \frac{\partial y_i}{\partial \phi_\gamma^j} &= \begin{cases} a_\gamma^j \cos(4\pi^2 \gamma f_j s(t) + \phi_\gamma^j), & i=j, \\ 0, & i \neq j, \end{cases} \end{aligned}$$

With  $i \in \{1, 2\}$ , for the derivative  $\nabla_\theta \hat{J}$ , we also need

$$\begin{aligned} c_i' &= \sum_{\gamma=1}^{\Gamma_i} 4\pi^2 \gamma f_i a_\gamma^i \cos(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i), \\ c_i'' &= - \sum_{\gamma=1}^{\Gamma_i} 16\pi^4 \gamma^2 f_i^2 a_\gamma^i \sin(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i). \end{aligned}$$

$$\begin{aligned} \frac{\partial c_i'}{\partial f_1} &= \begin{cases} \sum_{\gamma=1}^{\Gamma_i} (4\pi^2 \gamma a_\gamma^i \cos(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i) \\ - 16\pi^4 \gamma^2 a_\gamma^i \sin(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i)), & i=1, \\ 0, & i=2, \end{cases} \\ \frac{\partial c_i''}{\partial f_1} &= \begin{cases} \sum_{\gamma=1}^{\Gamma_i} (-32\pi^4 \gamma^2 a_\gamma^i \sin(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i) \\ - 64\pi^6 \gamma^3 a_\gamma^i \cos(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i)), & i=1, \\ 0, & i=2, \end{cases} \\ \frac{\partial c_i'}{\partial a_0^j} &= 0, \quad \frac{\partial c_i''}{\partial a_0^j} = 0, \\ \frac{\partial c_i'}{\partial a_\gamma^j} &= \begin{cases} \sum_{\gamma=1}^{\Gamma_i} 4\pi^2 \gamma f_i \cos(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i), & i=j, \\ 0, & i \neq j, \end{cases} \\ \frac{\partial c_i''}{\partial a_\gamma^j} &= \begin{cases} - \sum_{\gamma=1}^{\Gamma_i} 16\pi^4 \gamma^2 f_i^2 \sin(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i), & i=j, \\ 0, & i \neq j, \end{cases} \\ \frac{\partial c_i'}{\partial \phi_\gamma^j} &= \begin{cases} - \sum_{\gamma=1}^{\Gamma_i} 4\pi^2 \gamma f_i a_\gamma^i \sin(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i), & i=j, \\ 0, & i \neq j, \end{cases} \\ \frac{\partial c_i''}{\partial \phi_\gamma^j} &= \begin{cases} - \sum_{\gamma=1}^{\Gamma_i} 16\pi^4 \gamma^2 f_i^2 a_\gamma^i \cos(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i), & i=j, \\ 0, & i \neq j. \end{cases} \end{aligned}$$

When the robot moves along a curve with parameter vector  $\theta_-$  and detects an object at  $s_- = s(t)$ , for the initial parameter vector of the high-level OCP, for  $i \in \{1, 2\}$ , we set

$$\begin{aligned} \forall \gamma, \phi_\gamma^i &= 4\pi^2 \gamma f_i s_- + \phi_{\gamma-}^i, \\ a_0^i &= c_i(s_-, \theta_-) - \sum_{\gamma=1}^{\Gamma_i} a_{\gamma-}^i \sin(\phi_\gamma^i), \end{aligned}$$

such that only a small number of constraints of the high-level OCP is violated initially and Alg. 1 can converge fast.

## APPENDIX C: PROOFS

*Proof.* (Theorem 1) Since  $\lambda_2$  is the only costate that depends on  $u_1$  in (20), (22) implies that  $\lambda_2^*(t) u_{c,p,1}^*(t) \leq \lambda_2^*(t) u_{c,p,1}$  must hold. For  $\lambda_2(t) \neq 0, t \in [0, t_2)$ , we obtain  $u_{p,1}(t) = -\text{sgn}(\lambda_2(t))$ , i.e.  $u_1^*(t) \in \{\pm 1\}$ . As the input is bounded,  $\lambda_2(t) = 0$  can hold at isolated times, without violating (22). A singular arc may exist since  $\lambda_2(t) = 0$  can also hold over an interval. The necessary condition for this case is obtained from the generalized first-order Legendre-Clebsch condition [41]

$$\begin{aligned} \frac{d^2}{dt^2} \frac{\partial \tilde{H}}{\partial u_{c,p,1}} &= \ddot{\lambda}_2(t) = 2c_g \alpha m_0 \dot{\tilde{x}}_2 + \dot{\tilde{x}}_3 + \dot{\lambda}_1 = 0, \\ - \frac{\partial}{\partial u_{c,p,1}} \left[ \frac{d^2}{dt^2} \frac{\partial \tilde{H}}{\partial u_{c,p,1}} \right] &\neq 0, \end{aligned}$$

yielding  $-1/(2c_g)$  as an additional possible input value.  $\square$

*Proof.* (Theorem 2) From (20) and  $\tilde{x}_1 \geq 0$ ,  $\lambda_2(0) < 0$  holds and the optimal control starts with a fragment  $u_{c,p,1}^*(t) = 1$  for  $t \in [0, \bar{t}_1)$ . Assuming that  $\lambda_2(0) = a \in \mathbb{R}_{<0}$  for the interval  $[0, \bar{t}_1)$ , the adjoint variable is  $\lambda_2(t) = a - \frac{2c+1}{2} t^{(2)} - \lambda_1 t$  by integration of (21). If  $\lambda_2(\bar{t}_1) = \lambda_2(\bar{t}_1) = 0$ ,  $\lambda_1 = -\sqrt{2(1+2c)a}$ , the set of possible control sequences

is  $\{(1), (1, -1), (1, -1/2c_g), (1, -1/2c_g, -1)\}$ , or in generalized form

$$u_{c,p,1}(t) = \begin{cases} 1, & t \in [0, \bar{t}_1), \\ -\frac{1}{2c_g}, & t \in [\bar{t}_1, \bar{t}_2), \\ -1, & t \in [\bar{t}_2, t_2). \end{cases} \quad (40)$$

Integrating (2) with  $m_q = m_\emptyset$  and (40), using the switching and final conditions  $\tilde{x}_1(t_2) = 1, \tilde{x}_2(t_2) = 0, \tilde{x}_2(\bar{t}_2) = v_c$  and analyzing the expressions for  $\bar{t}_2 = \bar{t}_1$  and  $\bar{t}_2 = t_2$ , we obtain

$$\sqrt{\frac{2(m_\emptyset\alpha + c_g\alpha^2 m_\emptyset^2 v_c^2)}{2c_g + 1}} \leq \bar{t}_1 \leq \sqrt{m_\emptyset\alpha + \frac{m_\emptyset^2\alpha^2 v_c^2}{2}},$$

for  $0 \leq v_c \leq \sqrt{\frac{2}{m_\emptyset\alpha}}$ . Substituting the solutions of the ODE, the cost  $E\{\ell\}$  becomes a function of  $\bar{t}_1$  and  $v_c$ . Setting  $\frac{\partial E\{\ell\}}{\partial \bar{t}_1} = 0$ ,  $\bar{t}_1^* \in \left\{ \sqrt{\frac{2\alpha m_\emptyset}{2c_g + 1}}, \sqrt{m_\emptyset\alpha + \frac{\alpha^2 m_\emptyset^2 v_c^2}{2}} \right\}$  holds. Analyzing  $\frac{\partial^2 E\{\ell\}}{\partial \bar{t}_1^2}$  in the feasible interval, we can verify that the first value corresponds to the minimum. Substituting  $\bar{t}_1^*$  in  $E\{\ell\}$  and evaluating  $\frac{\partial E\{\ell\}}{\partial v_c}$ , for the optimal velocity  $v_c^* \in \left\{ 0, \sqrt{\frac{2}{m_\emptyset\alpha}} \right\}$  holds. Analyzing  $\frac{\partial^2 E\{\ell\}}{\partial v_c^2}$  yields  $v_c^* = 0$  and  $\bar{t}_2 = 2c_g \sqrt{\frac{2m_\emptyset\alpha}{2c_g + 1}}$ , leading to  $\bar{t}_2 = t_2$ . Solving the ODE with (40) yields the switching point  $1/(1 + 2c_g)$  and the corresponding controller.  $\square$

#### APPENDIX D: OPTIMAL CONTROL FOR TWO OBJECTS

First, consider the planned path shown in Fig. 3(a) with corresponding generalized trajectory shown in Fig. 3(c). Since  $o_2$  is discovered at latest when the robot reaches  $s_2$ ,  $\bar{x}_2(t_4) = 0$ . Thus, optimizing the trajectory for  $t \in [t_4, t_6]$  can be decoupled, and the optimal control is given by

$$u_c^{3*}(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x}_1 \in [s_2, \frac{1+s_2}{2}), \\ -1, & \text{if } \bar{x}_1 \in [\frac{1+s_2}{2}, 1). \end{cases}$$

To obtain the optimal switching times of the remaining trajectory, we solve the ODE (17) with

$$u_{c,p}^1(t) = \begin{cases} 1, & \text{if } t \in [0, t_1), \\ -1, & \text{if } t \in [t_1, t_2), \\ 1, & \text{if } t \in [t_2, t_3), \\ -\frac{1}{2c_g}, & \text{if } t \in [t_3, t_4). \end{cases}$$

With the boundary conditions  $\bar{x}_1(0) = \bar{x}_2(0) = 0$ , and  $\bar{x}_1(t_2) = s_1$ , we obtain

$$-\frac{(t_2 - t_1)^2}{2m} + \frac{t_1}{m}(t_2 - t_1) + \frac{t_1^2}{2m} = s_1,$$

where  $m = m_q\alpha$ , and since  $t_2 \geq t_1$ ,

$$t_1 = \begin{cases} t_2, & \text{if } t_2 \leq \sqrt{2ms_1}, \\ t_2 - \frac{\sqrt{2}}{2} \sqrt{t_2^2 - 2ms_1}, & \text{else.} \end{cases}$$

Using the boundary conditions  $\bar{x}_1(t_4) = s_2, \bar{x}_2(t_4) = 0$ , we obtain

$$-\frac{(t_4 - t_3)^2}{4c_g m} + \frac{t_3 + 2t_1 - 2t_2}{m}(t_4 - t_3) + \frac{t_3 - 3t_2 + 4t_1}{2m}(t_3 - t_2) + s_1 = s_2, \\ t_4 = (2c_g + 1)t_3 - 4c_g(t_2 - t_1),$$

yielding

$$t_3 = 2(t_2 - t_1) + \sqrt{\frac{t_2^2 - 4t_1 t_2 + 4t_1^2 + 2m(s_2 - s_1)}{2c_g + 1}}.$$

Since  $t_3 \in \mathbb{R}_+$  and  $t_3 \geq t_2$ , together with the derived expression for  $t_1$ , we obtain

$$t_3 = \begin{cases} \sqrt{\frac{t_2^2 + 2m(s_2 - s_1)}{2c_g + 1}}, & \text{if } t_1 = t_2, \\ t_2, & \text{else.} \end{cases}$$

From the continuity of the variables, there exists a time when  $t_1 = t_2 = t_3 = \sqrt{\frac{m}{c_g}}(s_2 - s_{sw})$ . Thus, the switching takes place at

$$s' = \frac{1}{2c_g + 1} s_2.$$

Note that this corresponds to the result of Theorem 2, scaled for the interval  $[0, s_2]$ . For  $t_2 > \sqrt{2ms_1}$ , substituting  $t_1$  into the expression for  $t_3 = t_2$ , we obtain  $t_2 = t_3 = \sqrt{\frac{m}{c_g}}(s_2 - s_1)$  and

$$t_{1,s} = \sqrt{\frac{m(s_2 - s_1)}{c_g}} - \frac{\sqrt{2}}{2} \sqrt{\frac{m(s_2 - s_1) - 2c_g m s_1}{c_g}}$$

yielding a switching at

$$s'' = -\frac{2^{\frac{3}{2}} \sqrt{s_2 - s_1} \sqrt{s_2 - (2c_g + 1)s_1} - 3s_2 + (2c_g + 3)s_1}{4c_g}$$

Thus, the optimal control is given by

$$u_{c,p}^{1*}(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x}_1 \in [0, s_{sw,1}), \\ -1, & \text{if } \bar{x}_1 \in [s_{sw,1}, s_{sw,2}), \\ -\frac{1}{2c_g}, & \text{if } \bar{x}_1 \in [s_{sw,2}, s_2), \end{cases} \quad (41)$$

where  $s_{sw,1} = \max\{s', s_1\}$  and

$$s_{sw,2} = \begin{cases} s'', & \text{if } s_1 > s', \\ s', & \text{else.} \end{cases}$$

Now consider the planned path shown in Fig. 3(b) with corresponding generalized trajectory shown in Fig. 3(d). Analogously to the above analysis, it can be easily shown that the optimal control is

$$u_{c,p}^{1*}(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x}_1 \in [0, s_{sw,1}), \\ -1, & \text{if } \bar{x}_1 \in [s_{sw,1}, s_{sw,2}), \\ -\frac{1}{2c_g}, & \text{if } \bar{x}_1 \in [s_{sw,2}, s_{sw,3}), \\ 1, & \text{if } \bar{x}_1 \in [s_{sw,3}, s_{sw,4}), \\ -1, & \text{if } \bar{x}_1 \in [s_{sw,4}, s_{sw,5}), \\ -\frac{1}{2c_g}, & \text{if } \bar{x}_1 \in [s_{sw,5}, s_2), \end{cases}$$

where with  $\bar{x}_2(t_4) = \bar{x}_2(t_6)$  we obtain the appropriate switching spots  $s_{sw,1}$  to  $s_{sw,5}$ . Thus, for  $n$  unexplored intervals in  $[0, 1]$ , it can be shown that the optimal control consists of a string of  $n$  controls of the form (41) with appropriate switching conditions.

## REFERENCES

- [1] S. Thrun, W. Burgard, and D. Fox., *Probabilistic Robotics*. MIT Press, 2008.
- [2] A. E. Bryson and Y. C. Ho, *Applied optimal control : optimization, estimation, and control*. John Wiley and Sons, New York, 1975.
- [3] F. Clarke, "Discontinuous feedback and nonlinear systems," in *Proc. of IFAC Symp. on Nonlinear Control Systems (NOLCOS)*, 2010.
- [4] D. Feng and B. H. Krogh, "Acceleration-constrained time-optimal control in  $n$  dimensions," *IEEE Trans. on Automatic Control*, vol. 31, no. 10, pp. 955–958, 1986.
- [5] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [6] P. Grieder, M. Kvasnica, M. Baotić, and M. Morari, "Stabilizing low complexity feedback control of constrained piecewise affine systems," *Automatica*, vol. 41, no. 10, pp. 1683 – 1694, 2005.
- [7] C. Seatzu, D. Corona, A. Giua, and A. Bemporad, "Optimal control of continuous-time switched affine systems," *IEEE Trans. on Automatic Control*, vol. 51, pp. 726–741, 2006.
- [8] V. Nenchev, C. Belta, and J. Raisch, "Optimal motion planning with temporal logic and switching constraints," in *14th European Control Conf. (ECC'15)*, Linz, Austria, 2015, pp. 1135–1140.
- [9] B. Passenberg and O. Stursberg, "Graph search for optimizing the discrete location sequence in hybrid optimal control," in *3rd IFAC Conf. on Analysis and Design of Hybrid Systems*, 2009, pp. 304–309.
- [10] C. G. Cassandras and J. Lygeros, Eds., *Stochastic hybrid systems*. CRC Press, 2010.
- [11] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, *The Traveling Salesman Problem: A Computational Study: A Computational Study*, ser. Princeton Series in Applied Mathematics. Princeton University Press, 2011.
- [12] R. Bellman, "An optimal search problem," *SIAM Review*, vol. 5, 1963.
- [13] L. J. Guibas, J.-C. Latombe, S. M. Lavalley, D. Lin, and R. Motwani, "A visibility-based pursuit-evasion problem," *International Journal of Computational Geometry and Applications*, vol. 9, pp. 471–494, 1996.
- [14] S. Alpern and S. Gal, *The Theory of Search Games and Rendezvous*. Kluwer's Int. Ser. in Oper. Research & Management Science, 2003.
- [15] M. Kao, J. Reif, and S. Tate, "Searching in an unknown environment: an optimal randomized algorithm for the cow-path problem," *Information and Computation*, vol. 131, no. 1, pp. 63–80, 1996.
- [16] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Robotics and Automation, IEEE Trans. on*, vol. 20, no. 2, pp. 243–255, 2004.
- [17] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," *IEEE Trans. on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.
- [18] J. Le Ny and G. Pappas, "On trajectory optimization for active sensing in gaussian process models," in *Proc. of 48th IEEE Conf. on Decision and Control (CDC) held jointly with 28th Chinese Control Conference (CCC)*, 2009, pp. 6286–6292.
- [19] S. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *Robotics, IEEE Trans. on*, vol. 28, no. 2, pp. 410–426, 2012.
- [20] C. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Trans. on Automatic Control*, vol. 58, no. 4, pp. 947–961, 2013.
- [21] A. Molin and S. Hirche, "On the optimality of certainty equivalence for event-triggered control systems," *IEEE Trans. on Automatic Control*, vol. 58, pp. 470–474, 2013.
- [22] H. Axelsson, M. Boccadoro, M. Egerstedt, P. Valigi, and Y. Wardi, "Optimal mode-switching for hybrid systems with varying initial states," *Nonlinear Analysis: Hybrid Systems*, vol. 2, no. 3, pp. 765–772, 2008.
- [23] M. Rickert, A. Sieverling, and O. Brock, "Balancing exploration and exploitation in sampling-based motion planning," *IEEE Trans. on Robotics*, vol. 30, no. 6, pp. 1305–1317, 2014.
- [24] M. Toussaint, "The bayesian search game," in *Theory and Principled Methods for Designing Metaheuristics*, ser. Natural Computing Series, Y. Borenstein and A. Moraglio, Eds. Springer, 2014, pp. 129–144.
- [25] C. Amato, G. Konidaris, A. Anders, G. Cruz, J. P. How, and L. P. Kaelbling, "Policy search for multi-robot coordination under uncertainty," in *Proc. of Robotics: Science and Systems Conf. (RSS-15)*, 2015.
- [26] M. Lahijanian, M. R. Maly, D. Fried, L. E. Kavraki, H. Kress-Gazit, and M. Y. Vardi, "Iterative temporal planning in uncertain environments with partial satisfaction guarantees," *IEEE Trans. on Robotics*, vol. 32, no. 3, pp. 583–599, 2016.
- [27] V. Raman, N. Piterman, C. Finucane, and H. Kress-Gazit, "Timing semantics for abstraction and execution of synthesized high-level robot control," *IEEE Trans. on Robotics*, vol. 31, no. 3, pp. 591–604, 2015.
- [28] K. S. Wesselowski and C. G. Cassandras, "The elevator dispatching problem: Hybrid system modeling and receding horizon control," in *Proc. of 2nd IFAC Conf. on Analysis and Design of Hybrid Systems*, 2006, pp. 136–141.
- [29] Y. Khazaeni and C. G. Cassandras, "A new event-driven cooperative receding horizon controller for multi-agent systems in uncertain environments," in *Proc. of 53rd Conf. on Decision and Control (CDC'14)*, 2014, pp. 2770–2775.
- [30] V. Nenchev and C. Belta, "Receding horizon robot control in uncertain environments with temporal logic constraints," in *15th European Control Conf. (ECC'16)*, 2016, pp. 2614–2619.
- [31] V. Nenchev and J. Raisch, "Towards time-optimal exploration and control by an autonomous robot," in *Proc. of 21st Mediterranean Conf. on Control and Automation (MED'13)*, 2013, p. 1236–1241.
- [32] V. Nenchev and C. G. Cassandras, "Optimal exploration and control for a robotic pick-up and delivery problem," in *Proc. of the 53rd Conf. on Decision and Control (CDC'14)*, 2014, pp. 7–12.
- [33] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, and C. Yao, "Perturbation analysis and optimization of stochastic hybrid systems," *European Journal of Control*, vol. 16, no. 6, pp. 642 – 661, 2010.
- [34] V. Nenchev and C. G. Cassandras, "Optimal exploration and control for a robotic pick-up and delivery problem in two dimensions," in *Proc. of the 54th Conf. on Decision and Control (CDC'15)*, 2015, pp. 258–263.
- [35] T. Henzinger, *The Theory of Hybrid Automata*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 265–292.
- [36] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *Automatic Control, IEEE Trans. on*, vol. 54, no. 10, pp. 2318–2327, 2009.
- [37] E. A. Gol, M. Lazar, and C. Belta, "Language-guided controller synthesis for linear systems," *IEEE Trans. on Automatic Control*, vol. 59, no. 5, pp. 1163–1176, 2014.
- [38] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Athena Scientific, 1996.
- [39] G. Bao and C. G. Cassandras, "Stochastic comparison algorithm for continuous optimization with estimation," *Journal of optimization theory and applications*, vol. 91, no. 3, pp. 585–615, 1996.
- [40] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [41] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. The MIT Press, 2005.